

# CS210 – Machine Problem 6

Zombie Voting Rights Act of 2017

20 Points

Assigned: Mar. 27-28, 2017

Due: Apr. 3-4, 2017

The 28th amendment to the constitution granted voting rights to the undead population. However, since zombies are not technically full humans, their vote only counts as a fraction of a full, human, vote. Every ten years, when a zombie registers for the census, he or she must submit the fraction of their original, living mass, that they currently possess as a ratio of whole pounds. Therefore, in order to preserve the franchise of the undead, it is critically important for our computing systems to deal correctly with fractions.<sup>1</sup>

Your task is to implement a library for handling rational number (fractions) arithmetic. For this project, you will not submit a `main.c` file, you will simply submit the `rational.c` and `rational.h` files in a single .zip file. I will run my driver program to test your implementation.

## Specifications

- The type of rational numbers should be called `rational_t`.
- It should have two integer components named `numerator` and `denominator`.
- You must provide a function `rational_t make_rational(int num, int den)` that takes two integer arguments representing the numerator and denominator of a fraction and returns a `rational_t`.
- You must provide a function `void r2a(rational_t x, char result[])` that puts a string representation of `x` into the string `result`. The format should be: `num/den` (with no spaces).
- You must provide functions for the four main arithmetic operations named `rat_add`, `rat_sub`, `rat_mul`, and `rat_div`. All of these should take two `rational_t` arguments and return a new `rational_t`. You may use the following formulae:

$$\frac{a}{b} + \frac{c}{d} = \frac{ad + bc}{bd}$$

$$\frac{a}{b} - \frac{c}{d} = \frac{ad - bc}{bd}$$

$$\frac{a}{b} \cdot \frac{c}{d} = \frac{ac}{bd}$$

$$\frac{a}{b} \div \frac{c}{d} = \frac{ad}{bc}$$

---

<sup>1</sup>Thanks to Edgar Francis Qualkenbush II for this idea

- All `rational_t` numbers returned by the above functions should be in simplest form. This can be accomplished by finding the greatest common denominator (GCD) of the numerator and denominator and then dividing them both by it. The gcd of two numbers,  $a \geq 0$  and  $b \geq 0$  can be calculated using the following recursive formula (Note: the **mod** operation is the % operation in C):

$$\text{gcd}(a, b) = \begin{cases} a & \text{if } b = 0 \\ \text{gcd}(b, a) & \text{if } a < b \\ \text{gcd}(b, a \bmod b) & \text{otherwise} \end{cases}$$

Also, if the numerator and denominator have the same sign, then they should both be made positive. If they have different signs, the numerator should be negative and the denominator should be positive.